WHAT IS CLAIMED IS:

1. A method for scheduling at least one data transfer for a plurality of input/output (I/O) devices, each I/O device having a direct memory access (DMA) controller and being associated with one or more network ports, the method comprising the steps of:

polling, from a device interface, the plurality of I/O devices to receive status inputs from the I/O devices;

selecting an I/O device to be serviced based at least in part on the status inputs;

storing a first identifier associated with the selected I/O device in a first register of the device interface;

accessing, at a processor, the first identifier from the first register of the device interface;

selecting a handler routine from a plurality of handler routines based at least in part on the first identifier; and

executing the selected handler routine at the processor to process a data transfer with the selected I/O device or DMA controller.

2. The method as in Claim 1, wherein the step of selecting an I/O device involves selecting a DMA controller associated with the I/O device.

3. The method as in Claim 2, wherein the first identifier is associated with the DMA controller of an I/O device.

4. The method as in Claim 2, wherein the data transfer processed by the handler routine includes a DMA transfer of incoming data from an I/O device to memory using the DMA controller.

5. The method as in Claim 2, wherein the data transfer processed by the handler routine includes a DMA transfer of outgoing data from memory to an I/O device using the DMA controller.

6. The method as in Claim 1, wherein the selected I/O device is an input device and the data transfer processed by the handler routine includes transferring data from the input device.

7. The method as in Claim 1, wherein the selected I/O device is an output device and the data transfer processed by the handler routine includes transferring data to the output device.

8. The method as in Claim 1, wherein references to the plurality of handler routines are arranged as a plurality of table entries of a table stored in memory accessible by the processor, each table entry having a context pointer referencing a data structure representing a state of operation of a respective I/O device or DMA controller and a handler routine address referencing a memory address of a corresponding handler routine and wherein the identifier is identifies a certain table entry.

9. The method as in Claim 8, wherein the step of executing the handler routine includes executing a handler routine referenced by the handler routine address of the identified table entry using the data structure referenced by the context pointer of the identified table entry.

10. The method as in Claim 1, wherein a status input from an I/O device includes one or more of: an indication of whether the I/O device requires servicing; an indication of which of the one or more network ports associated with the I/O device requires servicing; and an indication of a priority.

11. The method as in Claim 1, further comprising the steps of:

storing a second identifier representing a network port associated with the selected I/O device in a second register of the device interface; and

accessing, at the processor, the second identifier from the second register of the device interface; and

wherein the selected handler routine is executed using the second identifier.

12. A system for scheduling a data transfer for at least one of a plurality of input/output (I/O) devices, each I/O device having a direct memory access (DMA) controller and being associated with at least one network port, the system comprising:

a device interface operably connected to the plurality of I/O devices and being adapted to:

poll the plurality of I/O to receive status inputs from the I/O devices;

select an I/O device to be serviced based at least in part on the status inputs; and

store a first identifier associated with the selected I/O device in a first register of the device interface; and

a processor operably connected to the device interface and being adapted to:

access the first identifier from the first register of the device interface;

select a handler routine from a plurality of handler routines based at least in part on the first identifier; and

execute the selected handler routine to process a data transfer with the selected I/O device.

13. The system as in Claim 12, wherein selecting an I/O device involves selecting a DMA controller associated with the I/O device.

14. The system as in Claim 13, wherein the first identifier is associated with the DMA controller of an I/O device.

15. The system as in Claim 13, wherein the data transfer processed by the handler routine includes a DMA transfer of incoming data from an I/O device to memory using the DMA controller.

16. The system as in Claim 13, wherein the data transfer processed by the handler routine includes a DMA transfer of outgoing data from memory to an I/O device using the DMA controller.

17. The system as in Claim 12, wherein the selected I/O device is an input device and the data transfer processed by the handler routine includes transferring data from the input device.

18. The system as in Claim 12, wherein the selected I/O device is an output device and the data transfer processed by the handler routine includes transferring data to the output device.

19. The system as in Claim 12, wherein references to the plurality of handler routines are arranged as a plurality of table entries of a table stored in memory accessible to the processor, each table entry having a context pointer referencing a data structure representing a state of operation of the selected I/O device or DMA controller and a handler routine address referencing a memory address of a corresponding handler routine and wherein the identifier is identifies a certain table entry.

20. The system as in Claim 19, wherein executing the handler routine includes executing a handler routine referenced by a handler routine address of the identified table entry using a data structure referenced by a context pointer of the identified table entry.

21. The system as in Claim 12, wherein a status input from an I/O device or DMA controller includes one or more of: an indication of whether the I/O device or DMA controller requires servicing; an indication of which of the one or more

network ports associated with the I/O device requires servicing; and an indication of a priority.

22. The system as in Claim 12, wherein:

the device interface is further adapted to store a second identifier representing a network port associated with the selected I/O device in a second register of the device interface; and

the processor is further adapted to access the second identifier from the second register of the device interface;

wherein the processor executes the selected handler routine using the second identifier.

23. A communications processor comprising:

a plurality of input/output (I/O) devices, each I/O device comprising:

a direct memory access (DMA) controller; and

at least one network port;

a device interface operably connected to the plurality of I/O devices and having a first register, the device interface being adapted to:

poll the plurality of I/O devices to receive status inputs from the I/O devices and DMA controllers;

select an I/O device to be serviced based at least in part on the status inputs; and

store a first identifier associated with the selected I/O device in a first register of the device interface;

means for selecting a handler routine from a plurality of handler routines based at least in part on the first identifier; and

means for executing the selected handler routine to process a data transfer with the selected I/O device.

24. The communications processor as in Claim 23, wherein selecting an I/O device involves selecting a DMA controller associated with the I/O device.

25. The communications processor as in Claim 24, wherein the first identifier is associated with the DMA controller of an I/O device.

26. The communications processor as in Claim 24, wherein the data transfer processed by the handler routine includes a DMA transfer of incoming data from an I/O device to memory using the selected DMA controller.

27. The communications processor as in Claim 24, wherein the data transfer processed by the handler routine includes a DMA transfer of outgoing data from memory to an I/O device using the selected DMA controller.

28. The communications processor as in Claim 23, wherein the selected I/O device is an input device and the data transfer processed by the handler routine includes transferring data from the input device.

29. The communications processor as in Claim 23, wherein the selected I/O device is an output device and the data transfer processed by the handler routine includes transferring data to the output device.

30. The communications processor as in Claim 23, wherein references to the plurality of handler routines are arranged as a plurality of table entries of a table stored in memory, each table entry having a context pointer referencing a data structure representing a state of operation of the selected I/O device or DMA controller and a handler routine address referencing a memory address of a corresponding handler routine and wherein the identifier is identifies a certain table entry.

31. The communications processor as in Claim 30, wherein executing the handler routine includes executing a handler routine referenced by the handler routine address of the identified table entry using the data structure referenced by the context pointer of the identified table entry.

32. The communications processor as in Claim 23, wherein a status input from an I/O device or DMA controller includes one or more of: an indication of whether the I/O device or DMA controller requires servicing; an indication of which of the one or more network ports associated with the I/O device requires servicing; and an indication of a priority.

33. The communications processor as in Claim 23, wherein the device interface is further adapted to store a second identifier representing a network port associated with the selected I/O device or DMA controller in a second register of the device interface; and wherein the selected handler routine is executed using the second identifier.